A Normalized Mirrored Correlation Measure for Data Symmetry Detection

Alessandro Gnutti, Fabrizio Guerrini and Riccardo Leonardi
Department of Information Engineering
University of Brescia, Italy
Email: {a.gnutti006,fabrizio.guerrini,riccardo.leonardi}@unibs.it

Abstract—**Symmetry detection algorithms are enjoying a renovated interest in the scientific community, fueled by recent advancements in computer vision and computer graphics applications. This paper is inspired by recent efforts in building a symmetric object detection system in natural images. In particular, it is first shown how correlation can be a core operator that allows finding local reflection symmetry points in 1-D sequences that are optimal in an energetic sense. Then, the importance of 2-D correlation in natural images to correctly align the symmetric object axis is demonstrated. Using the correlation as described is crucial in boosting the performance of the system, as proven by the results on a standard dataset.**

Keywords–Normalized Cross-Correlation; Reflection Symmetry Detection; Content-Based Analysis; Feature Extraction Methods.

I. Introduction

Minimal distance between waveforms is known to maximize their cross-correlation. In digital communication, optimal receiver characteristics correspond to looking for the maximum crosscorrelation between the ideal noise-free candidate waveforms associated to the transmitted symbol and the actual received signal [1]. As another example, peaks in the normalized cross-correlation between a 2-D patch and a reference frame lead to finding the optimal shift that the patch has undergone in time describing its apparent motion, a major component used for redundancy reduction for digital transmission of digital video [2]. None of these methods however have considered the effect of mirrored versions of the waveform/patch which are instead clearly present in many natural signals (whether 1-D or multidimensional) as key patterns. This paper describes how correlation between a waveform and its mirrored version is a key concept to be considered.

In particular this paper explores the crucial role taken by correlated data in the workflow of a system proposed for symmetric object detection in natural images. We refer here to the most common symmetry found in both natural and manmade objects in natural images, the reflection symmetry. This is the most interesting of the symmetry kinds, which comprise translational and rotational symmetry as well, since it is well established that this is the most prevalent attribute in human visual perception (see [3] and references therein). From a physiological point of view, it is clear that much processes takes place in the cerebral cortex during object recognition tasks, and that symmetry detection is an important one of them [4] [5].

Computer vision researchers have recently attempted to replicate the brain efforts through the design of a suitable algorithm for symmetric objects detection. Such an interest culminated with a pair of dedicated workshops at the CVPR conference in 2011 and 2013, reporting the results of a competition for symmetric objects detection. In the 2013 iteration of the competition [6], two methods gave the best results. The first method [7] exhibits best results in single and multiple symmetry axis detection with recall rates lower than approximately 80%. It is based on a rotation invariant feature like SIFT descriptors, which create constellations of interest points that are finally matched. The second method [8] reports slightly better results in presence of high recall ratios. It uses SIFT descriptors in combination with gradient-based weighting to choose symmetry candidates that are then validated using a principled statistical approach. In the end, the results of the competition are still far from good performance. Looking for symmetric objects in natural images has even been proposed as a Turing test alternative to classic character-based queries [9]. Recent advances such as [10] have since improved the previously reported results on the CVPR 2013 competition database. This latter method, in particular, extracts robust, multi-scale 4-D Appearance of Structure descriptors computed through the detection of important local edges.

As it is well established in pattern recognition whether in 1-D or higher dimensions, correlation has a preeminent role, both in still images and in video, particularly in the techniques collectively known as appearance-based matching [11]. In those applications, the objective is to correlate some object characteristics, more commonly features extracted from the target object, with those possibly present in the considered content. The object detection works by correlating, in a wide sense, the target object features, under the prescribed application-dependent constraints, like scale- and rotation-invariance and robustness to occlusions, with those extracted from the content. This problem encompasses a number of design challenges, such as feature selection, classification algorithms and so on. However, in general there is no particular explicit interest in the correlation present in the features and/or data themselves to match the target object template to the corresponding object besides the implicit reduction in the information needed to represent the object.

The approach followed by in this paper is different, partially driven by the application aimed at by the proposed system. In the case at hand, there is no target object(s) to detect in a given image. Instead, we want to assess whether a symmetric object, whatever it may be, is indeed present. The only quality of the objects we are interested in is their symmetric appearance, so there is no way in which template-matching techniques can work in this case. It turns out that 2-D correlation is a crucial characteristic to tie what is detected merely as a symmetric signal to an actual symmetric object detection.

The question on how to exploit the particular correlation displayed by reflection symmetric data is tackled in two separate stages by the symmetric objects detection method. First,

ISBN 978-0-9928626-7-1 © EURASIP 2017

we cast the symmetry detection problem in the 1-D signals scenario and prove that correlation is at the core of its solution. Then, we move on to the search for the best local symmetry, still for 1-D signals, to prepare for the intended application. Last we extend the 1-D approach to 2-D, namely natural images. Here, correlation takes another, separate role in the success of the application. In particular, 2-D spatial correlation, which is inherent in the images content, helps to correct for errors in the symmetry placement as well as reinforce belief in the symmetry presence since it has a denoising effect on inconsistent symmetries found in adjacent rows.

The rest of the paper is organized as follows. In Section II the role of the correlation is considered for the detection of local symmetry points for 1-D sequences, in particular tying the concept of an optimal symmetry point found through the even/odd decomposition of a given sequence with the search for local maxima through the cross-correlation of a windowed sequence and its flipped version, namely leading to a self-convolution operation. In Section III instead the peculiar role of 2-D correlation is considered for handling the natural images domain, in particular showing how it allows to precisely detect main symmetry axes of symmetric objects. Conclusions are finally drawn in Section IV.

II. Correlation for symmetry detection in 1-D signals

This work starts its analysis on the nature of correlation in 1-D discrete data for symmetry detection. This Section highlights the main takeaway in this domain, that is the use of the basic even/odd signal basic decomposition. Thus, it is shown in an analytic way that correlating a windowed sequence with its mirrored version is the optimal approach to detect a reflection symmetry in 1-D data, which is also a satisfyingly intuitive result. Then, in the next Section the analysis is extended to the 2-D domain, and shows how 2-D spatially correlated data in images help in detecting symmetric objects as well.

Let us consider a real, finite-energy, 1-D discrete sequence $x[n]$, supposing for simplicity sake that it has a finite support. The even/odd decomposition states that $x[n]$ can be expressed as the sum of an even sequence $x_e[n]$ such that $x_e[n] = x_e[-n]$ and an odd sequence $x_o[n]$ such that $x_o[n] = -x_o[-n]$, as follows:

$$x_e[n] = \frac{x[n] + x[-n]}{2}; \quad x_o[n] = \frac{x[n] - x[-n]}{2} \tag{1}$$

and $x[n] = x_e[n] + x_o[n]$. It is readily observable that $x_e[n]$ and $x_o[n]$ are orthogonal in the sense that their inner (or scalar) product that is defined as $< x_e[n], x_o[n] >= \sum_n x_e[n] x_o^*[n]$ is 0. This also implies that the energies of $x_e[n]$ and $x_o[n]$, respectively E_e and E_o, sum up to the energy E of $x[n]$. Of course, this suggests a simple approach to determine how much (anti)-symmetric a sequence is around the time origin $n = 0$. After decomposing the signal $x[n]$ along the lines of Eq. (1), if $x[n]$ is prevalently even (resp. odd) the energy of the even sequence $x_e[n]$ is more (resp. less) than half of E, so comparing the latter sequences' energies is sufficient to hint at the symmetry properties of $x[n]$.

It is possible to generalize Eq. (1) in such a way to encompass also symmetries whose center is not in the time origin $n = 0$. To do that, it is necessary to flip the sequence around $n = m$ and Eq. (1) thus become:

$$x_e[n;m] = \frac{x[n] + x[2m-n]}{2}; \; x_o[n;m] = \frac{x[n] - x[2m-n]}{2} \tag{2}$$

and $x[n] = x_e[n;m] + x_o[n;m]$. Now, the objective is to evaluate the time instant where the sequence is maximally either even or odd. Focus is on the even case, since this addresses the issue of reflection symmetry detection. So, studying the energy $E_e(m)$ of the even sequence $x_e[n;m]$ as a function of the symmetry center m we have:

$$\begin{aligned} E_e(m) &= \sum_n |x_e[n;m]|^2 = \sum_n \left| \frac{x[n] + x[2m-n]}{2} \right|^2 = \\ &= \frac{1}{4} \sum_n |x[n]|^2 + |x[2m-n]|^2 + 2x[n]x[2m-n] = \\ &= \frac{1}{2} E + \frac{1}{2} \sum_n x[n]x[2m-n] = \frac{1}{2} E + (x * x)[2m] \end{aligned} \tag{3}$$

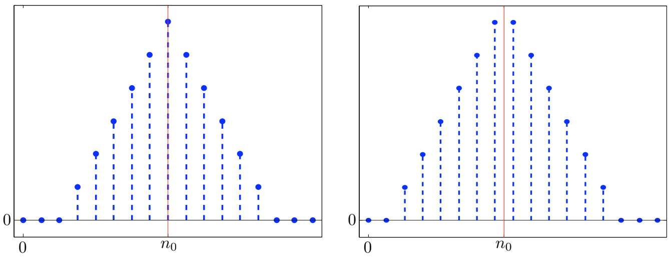
where in the last passage $(x * x)$ represents the convolution of the discrete sequence $x[n]$ with itself (termed in the following as "auto-convolution"). Thus, the optimal even symmetry point n_0 satisfies the condition:

$$2n_0 = \arg\max_m \; (x * x)[m] \tag{4}$$

Therefore, n_0 is found by analyzing where is the maximum of the auto-convolution of $x[n]$. As a matter of fact, as the factor 2 in Eq. (4) suggests, the globally optimal symmetry point is determined with a half-pixel accuracy. It is in fact possible that a perfectly even signal is symmetric around a half-integer folding point: Fig. 1 depicts with a very simple example the two types of even symmetries that can be defined for discrete-time sequences.

Finding n_0 as in Eq. (4) yields the globally optimal (even) symmetry point, in the sense that, among all the time instants around which to evaluate the energy of the even sequence, choosing n_0 guarantees that $E_e(n_0)$ is the maximum. That however does not mean that $x[n]$ possesses an even symmetry centered around n_0. As a matter of fact, all the other local maxima of the auto-convolution do not necessarily correspond to (possibly minor) local symmetries as well.

To explain why the correspondence between local symmetries and extreme points of the auto-convolution is not straightforward, let us analyze the simplified example case depicted in Fig. 2. The signal $x[n]$ of Fig. 2a has a couple



(a) Case with integer n_0.

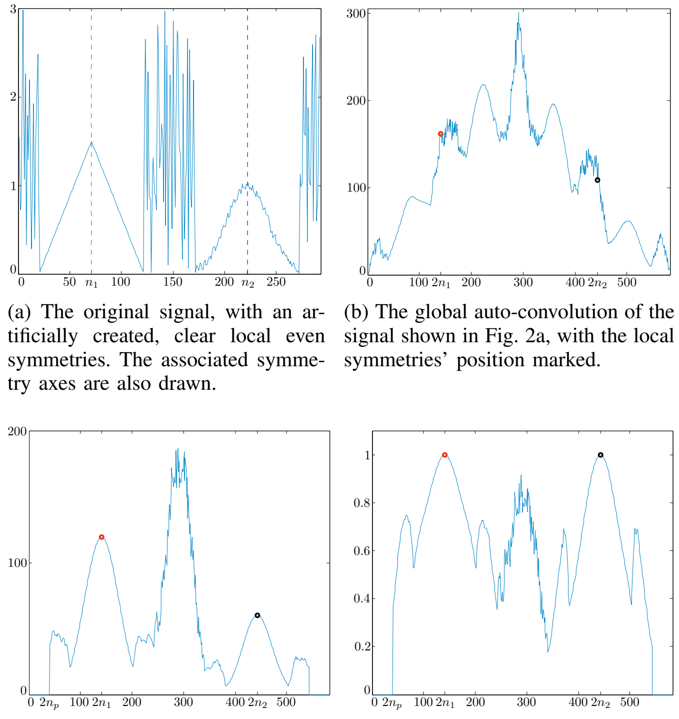
(b) Case with half-integer n_0.

Figure 1: Comparison of two even signals displaying resp. integer and half-integer optimal symmetry points.

of very clear even symmetries that are easy to spot, a left one where a perfectly triangular impulse is located and a right one with an approximately Gaussian impulse. The dotted red and black lines in Fig. 2a represent the vertical symmetry axes associated with the ideal symmetry points n_1 and n_2. These local symmetries are immersed in white noise having comparable energy, which may even be locally symmetric by chance but obviously in a far less obvious way than the impulses. What would ideally be expected is to find two extreme points in correspondence to the symmetric impulses (and not one in between since they have different shape).

In Fig. 2b we have reported the auto-convolution computed on the whole sequence, as dictated by Eq. (3). The red and black circles illustrate the position of $2n_1$ and $2n_2$ (recall that in Eq. (4) there is a multiplicative factor 2), corresponding to the desired symmetry points. The maximum of the auto-convolution is nowhere near those positions, and no local maxima are even associated to the desired locations. This happens because the auto-convolution value computed near the center of the symmetric impulses also depends on the non-symmetric portion of the signal as well as other symmetric impulses. In the end, the clearly perceived symmetries in Fig. 2a are not detectable in Fig. 2b.

Therefore, it is clear that the noise outside the symmetric parts of the signal must be neutered. To do this, a possible solution is to employ "windows", that is to limit the extent of the convolution support computed for a given symmetry center. This process may at first look similar to the one in [12]. However, in that work the search for local symmetry still took inspiration from a reasoning similar to that conveyed by Eq. (4) but actually performed in a totally different way. There, the overall auto-convolution were computed and then all local extrema were considered as starting candidate symmetry points. These candidates were then ranked by computing the even/odd decomposition, and the consequent energy decoupling, in increasingly larger windows around them as a validation process. This entire process was devised to both filter out those maxima that are false positives caused by noise and to compensate for the imprecise symmetry point location, which however is not corrected by the validation process.



(a) The original signal, with an artificially created, clear local even symmetries. The associated symmetry axes are also drawn.

(b) The global auto-convolution of the signal shown in Fig. 2a, with the local symmetries' position marked.

(c) Windowed auto-convolution of the signal shown in Fig. 2a. A high energy, weaker symmetry can dominate a low energy, stronger one due to the lack of energy normalization.

(d) Windowed and energy normalized auto-convolution of the signal of Fig. 2a. Now, the correct even symmetries are dominant.

Figure 2: The effect of windowing and normalization of the auto-convolution to improve the search for local symmetry. The borders are set to 0 (where windows go out of bounds).

In this paper, the auto-convolution is computed on a local window from the start. This way, the part of the signal that is not contained in the window support centered around the considered location can not influence the auto-convolution computation. In Fig. 2c the effect of the windowing is pretty clear since now the red and black circles clearly mark two extreme points. Still, the detected symmetry, namely the maximum of the windowed auto-convolution, is offset towards the center of the signal. In the end, even if the found optimal symmetry is not as discernible as the desired local ones, it shows a higher auto-convolution value because the energy of the windowed sequence is higher. This problem can be alleviated by normalizing the windowed auto-convolution by the energy of the sequence contained in the window. Therefore, the resulting windowed and normalized auto-convolution is illustrated in Fig. 2d. Of course, the value is now confined in the $[0, 1]$ interval. The red and black circles now correctly identify the two main local symmetries exhibited by Fig. 2a.

The extent of the window is of course an important parameter that may be adjusted as needed using a procedure similar to that adopted in [12]. However, to keep the discussion simple, for the rest of the paper fixed-length windows covering $n_p = 100$ locations around each extreme point are employed. This window size is a good tradeoff between keeping low the number of false positives while still achieving sufficient denoising in the auto-convolution computation.

To summarize, computing the auto-convolution considering just the portion of the signal belonging to a window centered on a given location is in effect equivalent to computing the cross-correlation of the windowed sequence and its mirrored version. This is also quite satisfying as an intuitive insight: the best local symmetry point is the one where the windowed sequence correlates the best with its flipped version. Note how, using windows, it is not necessary to search for the maximum of the windowed auto-convolution, since that would produce the optimal symmetry point of the window. Instead, we can just slide the window and then compare the above cross-correlation among all the possible window center locations. These cross-correlation values must be normalized by the windowed energy sequence to make the comparison of the symmetry extent possible. We can thus define a measure of symmetry $S(W)$ associated to a given window W that covers $2n_p+1$ positions, as follows:

$$S(W) = \frac{\sum_{n=-n_p}^{n_p} x[n] \cdot x[-n]}{\sum_{n=-n_p}^{n_p} |x[n]|^2} \tag{5}$$

Recall that Eq. (4) outputs the globally optimal symmetry point with half-pixel resolution. If we want to retain the same resolution even after windowing the auto-convolution as in Eq. (5), it follows that the window should be centered at half-integer positions. If the computation is confined to the original samples $x[n]$, it may be a problem to keep the number of positions inside W invariant when centered in either integer or half-integer positions. The simplest solution is to interpolate $x[n]$ by a factor of 2, doubling n_p and compute $S(W)$ on the interpolated image, so that such computation uses consistent values for adjacent symmetry positions.

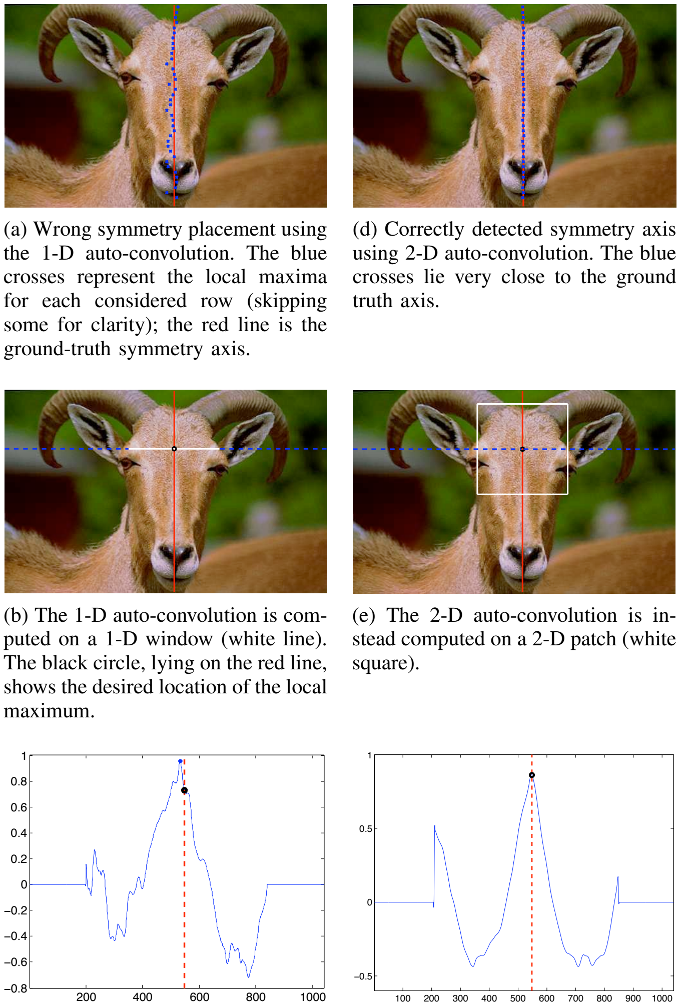
III. Correlation for symmetric object detection in natural images

As described in Section II, the cross-correlation between a windowed 1-D sequence and its mirrored version, whose window center is shifted across the signal, allows to locate the position of the optimal local symmetry. In this Section, we extend the method for the identification of the main local symmetry axes in a given image I, thus hopefully identifying symmetric objects therein. The first step towards the main objective is to compute a symmetry value for each pixel of I and for each direction of the symmetry. Regarding the axes direction, however, we will limit the discussion to the vertical axis since the method can be promptly applied to any direction by first suitably rotating the image. Therefore, searching for symmetric objects means to analyze the rotated image along its rows, however for the vertical axis direction considered in this paper there is no need to rotate the original image I.

One straightforward extension of the 1-D search for local symmetric points would be to apply Eq. (5) on each row of the image and then connect found maxima on different rows to identify the symmetric axis of the symmetric object. Thus, it could be expected that for all the rows associated to the symmetric object a local maximum is to be found where the symmetric object axis is placed. If this would be the case, simply connecting the extreme points in each row along the vertical direction would output the vertical symmetry object axis. This is actually a sound strategy as opposed to just taking the symmetry value. In fact, searching for the local maxima of the symmetry values is more important than the absolute value of the coefficient itself since its relationship with respect to its neighboring coefficients can correctly place the center of the symmetric pattern regardless of the nature of the pattern. A flood-fill algorithm can then be applied to connect the extreme points and thus all possible reflection symmetries are defined by a set of segments that join the local maxima found in adjacent rows.

However, Fig. 3 shows how such an approach does not work on natural images with an example. In particular, the left column of Fig. 3 depicts the found local maxima when separately analyzing the rows of the image. Fig. 3a shows that the detected points (blue crosses) do not always correspond to the symmetric object axis (red line), thus in the end the detected axis may be broken up or missed altogether. In those rows where the symmetry is off center the illumination and the textures combine to shift the 1-D local symmetry in the data. For example, let us analyze the row highlighted in blue in Fig. 3b. The black circle represents the ground-truth symmetry location and the 1-D window is depicted with the white line, when the window is indeed centered in the black circle's position. The sliding auto-correlation value given by Eq. (5) is plotted in Fig. 3c, and the maximum (the blue cross which corresponds to the one in the same row of Fig. 3b) is actually to the left of the desired location. This is a case where the true symmetry of the 1-D data and the perceived symmetry differ, in this case due to noise and local shadows, but that could also be caused by clutter and partial occlusions of the object.

However, our brain is able to effectively use the reflection symmetry information in the whole 2-D neighborhood, in effect exploiting the 2-D correlation in the image data. So, when perceiving the symmetric object the fact that the positioning of the actual symmetry in each 1-D row is imprecise is not important, in fact our brain can easily spot the vertical symmetry and extend it all the way. To adequately mimic such process, we propose to simultaneously compute a number of 1-D correlations in adjacent rows. In the end, what is actually computed for a given position to measure its level of symmetry is a windowed and normalized auto-convolution as before, but integrated on a 2-D square patch along the vertical direction



(a) Wrong symmetry placement using the 1-D auto-convolution. The blue crosses represent the local maxima for each considered row (skipping some for clarity); the red line is the ground-truth symmetry axis.

(d) Correctly detected symmetry axis using 2-D auto-convolution. The blue crosses lie very close to the ground truth axis.

(b) The 1-D auto-convolution is computed on a 1-D window (white line). The black circle, lying on the red line, shows the desired location of the local maximum.

(e) The 2-D auto-convolution is instead computed on a 2-D patch (white square).

(c) The 1-D auto-convolution computed along the blue line in Fig. 3b.

(f) The 2-D auto-convolution computed along the blue line in Fig. 3e.

Figure 3: Examples of a vertical symmetry axis detection, showing how computing the auto-convolution simultaneously on a 2-D patch instead of separately for each 1-D row permits to significantly correct the location of the symmetry.

ISBN 978-0-9928626-7-1 © EURASIP 2017

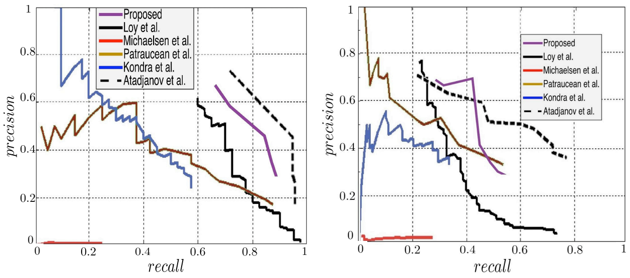
instead of on a 1-D window. So, instead of computing $S(W)$ as in Eq. (5), we compute $S_2(P)$ for any position in the center of the patch P:

$$S_2(P) = \frac{\sum_{m=-m_p}^{m_p} \sum_{n=-n_p}^{n_p} x[m,n] \cdot x[m,-n]}{\sum_{m=-n_p}^{n_p} \sum_{n=-n_p}^{n_p} |x[m,n]|^2} \qquad (6)$$

Computing the symmetry measure on the 2-D patch is in the end very beneficial to accurately place the symmetry location, as it is shown in the right column of Fig. 3. In Fig. 3d the almost perfect alignment of the detected symmetries to the ground-truth symmetry axis is depicted. Fig. 3e again exemplifies the process with the same row (in blue) considered in Fig. 3b, but this time the symmetry measure S_2, for the ground-truth symmetry location represented by the black circle, is computed on the 2-D patch drawn in white according to Eq. (6). Looking at the evolution of such measure along the row positions, as illustrated in Fig. 3f, it can be seen that the maximum is perfectly placed, *i.e.* the black circle is superimposed to the blue cross in the same row of Fig. 3e. In this case, therefore, the local symmetry is correctly tracked.

So, to summarize there are two key factors to achieve good performance of this application in natural images that are tightly connected with the concept of 2-D vertical correlation. First, the computation of the symmetry measure with respect to an entire 2-D patch centered around any one pixel location to compensate for noise or slight changes in the symmetric appearance: the 2-D neighborhood correlation (in presence of a symmetry) will boost a local maximum in the center of the patch. In addition, the presence of a local maximum is sufficient regardless of its value, *i.e.* there is no need for local thresholding, given that the integration effect of the 2-D patch will avoid any fluctuation of the symmetry measure. The second pivotal component is the validation of the consistency of the local maxima through the connectivity analysis carried out along each expected orientation of symmetry, in effect extending the 2-D vertical correlation to neighboring patches. This reinforces the belief that a symmetry is present while also allowing to define the extent of each symmetry segment. These important characteristics that depend on correlation allow to achieve very good performance in terms of precision-recall curves on the 2013 CVPR competition dataset, that are reported by Fig. 4. Details on the experimental procedure, that comprise both single and multiple symmetry axes detection scenarios, can be found in [6].

The computational complexity of the method is fairly low: on a standard desktop computer, the analysis of the symmetry axes on each angle for a 256 × 256 image takes on average slightly more than 1s. We end by noting that the result is robust to slight changes of the patch size (as we mentioned, in the conducted experiments reference size was set to $n_p = 100$). The patch size needs to represent only relevant data (which means confined to the extent of the symmetry), avoiding losing the coherence of local maxima. It can easily be expected that an appropriate multiresolution approach (image pyramid) would allow to find hierarchies of symmetries in the data.



(a) Single axis test scenario.

(b) Multiple axes test scenario.

Figure 4: Precision-recall curve of the proposed algorithm on the competition dataset compared with those in [6] and [10].

IV. Conclusions

In this paper we explored the role of correlation for the identification of data symmetries. First, it is analytically derived how correlation in 1-D sequences allows to find local reflection (*i.e.* even) symmetry points, in particular by computing the cross-correlation between the signal confined in a window and its mirrored version, normalized by the energy of the windowed signal. Then, the approach is extended to the search for symmetry in natural images, proving that naively applying the 1-D correlation is insufficient. Instead, exploiting the 2-D correlation in a square patch permits to correctly place the symmetry axis and also to reinforce belief in its presence. Furthermore, by joining detected symmetry points with a flood-fill algorithm also extends the 2-D correlation across adjacent patches. In the end, correlation in both the direction of the symmetry axis and the one orthogonal to it can be effectively used to boost performance, regardless of any particular feature being used.

References

[1] B. Sklar, Digital communications. Prentice Hall Upper Saddle River, 2001, vol. 2.

[2] I. E. Richardson, H. 264 and MPEG-4 video compression: video coding for next-generation multimedia. John Wiley & Sons, 2004.

[3] J. Wagemans, “Characteristics and models of human symmetry detection,” Trends in cognitive sciences, vol. 1, no. 9, 1997, pp. 346–352.

[4] J. DiCarlo, D. Zoccolan, and N. Rust, “How does the brain solve visual object recognition?” Neuron, vol. 73, no. 3, 2012, pp. 415–434.

[5] K. Grill-Spector, Z. Kourtzi, and N. Kanwisher, “The lateral occipital complex and its role in object recognition,” Vision research, vol. 41, no. 10, 2001, pp. 1409–1422.

[6] J. Liu, G. Slota, G. Zheng, Z. Wu, M. Park, S. Lee, I. Rauschert, and Y. Liu, “Symmetry detection from realworld images competition 2013: Summary and results,” in Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) Workshops, 2013, pp. 200–205.

[7] G. Loy and J. Eklundh, “Detecting symmetry and symmetric constellations of features.” in Proc. European Conf. on Computer Vision (ECCV), ser. Lecture Notes in Computer Science, A. Leonardis, H. Bischof, and A. Pinz, Eds., vol. 3952. Springer, 2006, pp. 508–521.

[8] V. Patraucean, R. Grompone von Gioi, and M. Ovsjanikov, “Detection of mirror-symmetric image patches,” in Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) Workshops, 2013, pp. 211–216.

[9] C. Funk and Y. Liu, “Symmetry recaptcha,” in Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 5165–5174.

[10] I. Atadjanov and S. Lee, “Reflection symmetry detection via appearance of structure descriptor,” in Proc. European Conf. on Computer Vision (ECCV), 2016, pp. 3–18.

[11] P. M. Roth and M. Winter, “Survey of appearance-based methods for object recognition,” Inst. for Computer Graphics and Vision, Graz University of Technology, Austria, Technical Report ICGTR0108 (ICG-TR-01/08), 2008.

[12] A. Gnutti, F. Guerrini, and R. Leonardi, “Representation of signals by local symmetry decomposition,” in Proc. European Signal Processing Conf. (EUSIPCO), 2015, pp. 983–987.